SETSCI conference proceedings
SETSCI Conference Proceedings
14 ,34-37, 2022

6th International Symposium on Innovative Approaches in Smart Technologies
December 8-10, 2022, Turkey

ISAS WINTER-2022

https://doi.org/10.36287/setsci.5.2.007

CNN HYPERPARAMETERS OPTIMIZATION USING RANDOM SEARCH FOR IMAGE CLASSIFICATION

Dala KRAYEM, Asmaallah FALLAHA, Mohamad Taj Eddin ASHOUR, Saed ALQARALEH*
Computer Engineering Department, Faculty of Engineering, Hassan Kalyoncu University, Gaziantep-TURKEY
saed.alqaraleh@hku.edu.tr

Abstract – In the last decade, deep learning in general and Convolutional Neural Networks (CNN) have achieved a human level and outstanding performance in almost all computer-based systems such as classification(text, images, and videos). Building an efficient CNN architecture and finding the most suitable layers requires time. In addition, tuning and setting CNN parameters such as the number of filters, size of filters, Dense rate, Dropout value, etc., has an essential effect on the overall performance of the CNN model.
This paper investigated the effects of CNN hyperparameter optimizations using Random Search. To achieve our goals, we selected to investigate the hyperparameter optimizations of MobileNetV2 and VGG19 models when used for image classification. Here, we have combined four mask detection datasets of approximately 6K, 12K, 4k, and 4k images into one dataset.
Overall, the results of the experimental works showed that using the hyperparameter optimization technique improved the overall performance of both MobileNetV2 and VGG19 models. In addition, VGG19 outperformed MobileNetV2 across Accuracy, Precision, recall, and F1 score evaluation matrixes.

Keywords – CNN, Deep learning, image classification, random search, hyperparameter optimization, Neural Architecture Search.

I. INTRODUCTION

Convolutional neural networks, also known as (CNN) have defied expectations and ascended to the throne as the most advanced computer vision amongst other classification algorithms varieties in the last few years. In the world of image data, these convolutional neural network models are widely used on computer vision tasks like image classification and perform phenomenally well.

The CNN architecture includes several building blocks; Convolution layers, pooling layers, Dropout layers, activation function layers, flatten layers, and fully connected layers, etc. The recurrence of a stack of numerous convolution layers and a pooling layer, followed by one or more fully connected layers, makes up a typical design of the CNN model.

The goal of the CNN model in the convolution layer is to determine which kernels perform best. However, the size of the kernels, the number of kernels, the padding, and the stride are hyperparameters that play a significant role in how well a model performs. Dropout value and Dense rate in Dense and Dropout layers also have a crucial impact on the overall performance of the CNN model. This is why we took advantage of hyperparameter optimization techniques to find the optimal hyperparameters.

There are some common strategies for optimizing hyperparameters, such as Grid search. It operates by attempting each and every conceivable set of the model's parameters. This implies that the full search will take a long time and is computationally expensive. Another strategy is Random search which is the approach used in this study. Random search works by testing several combinations that are selected randomly, as its name suggests. Because this approach tests fewer combinations, it calls for less computational time for the results to be obtained. The Genetic Algorithm is an advanced technique that employs natural processes like mutation, evolution, reproduction, etc. These algorithms employ sophisticated arithmetic computations but operate according to natural laws. It is possible to simulate the growth of an organism in its environment and how it interacts with other creatures using straightforward mathematical equations.

In this paper, we try to determine the best neural network design for MobileNetV2 and VGG19 using the Random search hyperparameter optimization approach.

The topic of hyperparameter optimization has recently attracted researchers' attention, and some of these recent studies are summarized below. In [1], the execution time of Grid Search, Random Search, and Genetic Algorithm was compared for hyperparameter optimization. The CIFAR-10 dataset was used in the experiments of [1], and the results indicated that the genetic algorithm is the fastest, especially when we try to optimize a large number of parameters.

In [2], a new hyperparameter optimization method for Neural architecture search (NAS) was introduced. This method is based on an improved version of early-stopping random search, which uses a weight-sharing algorithm. Next, the performance of random search with early-stopping was compared with the proposed method using PTB and CIFAR-

10 datasets, and the results of [2] showed that random search with early-stopping has been outperformed by random search with weight-sharing.

A possible solution for the problem of the slow and expensive execution of Neural Architecture Search (NAS), i.e., hyperparameter optimization techniques, was introduced in [3]. Mainly, it is based on the overlap of activations in untrained networks. In other words, this method searches for the optimal values without training the used model. Results on NAS-Bench-101, NAS-Bench-201, NATS-Bench datasets, and Network Design Spaces indicated that this approach could be efficiently integrated with more expensive search methods.

II. IMAGE CLASSIFICATION SYSTEMS

Image classification is a complex procedure of extracting context from an image by giving it a label depending on what the image is, and this process relies on various factors.

The main processes(steps) of image classification can be shown in the following figure(Figure 1), and the detail of these steps is summarized below.

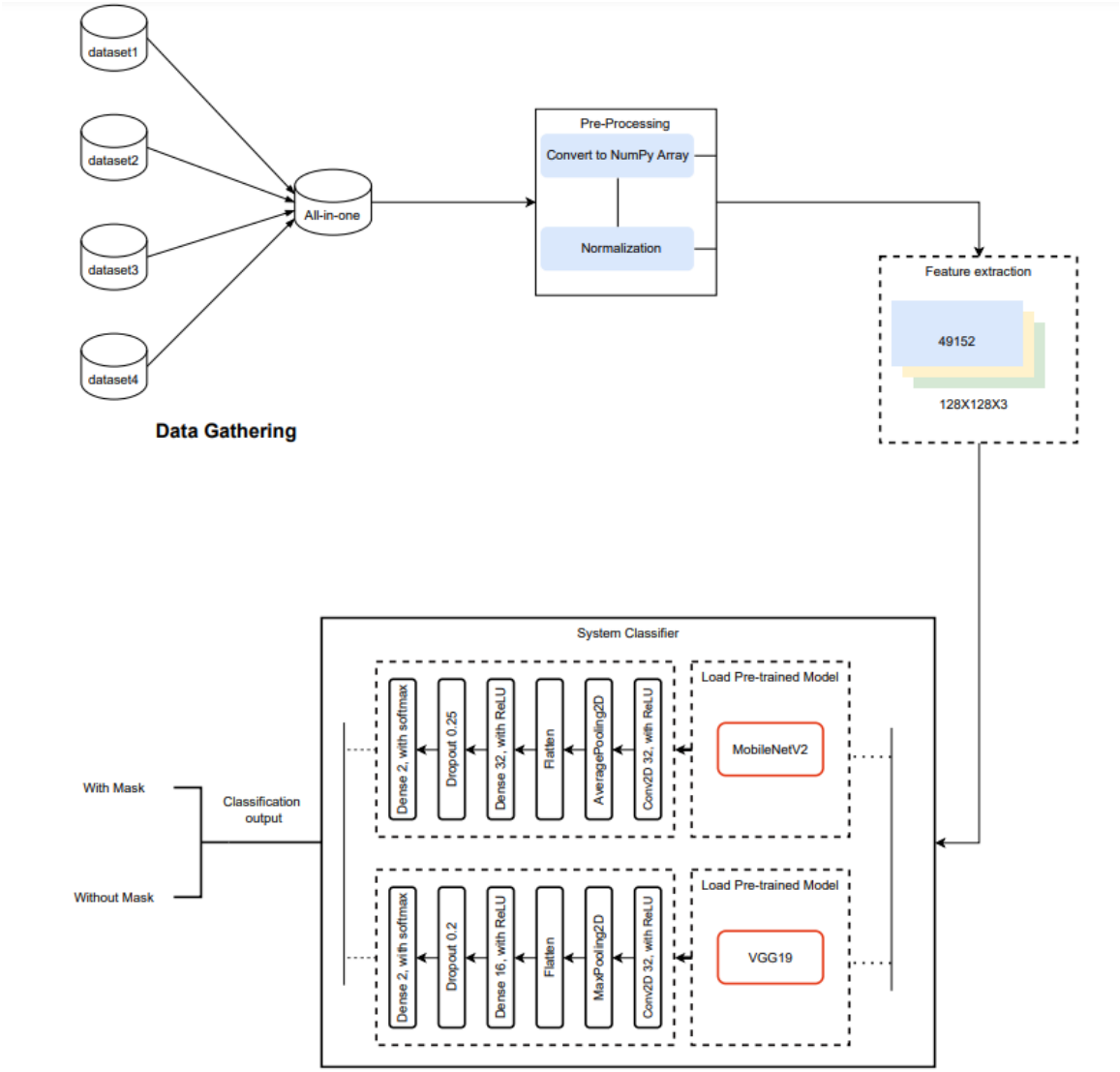

Fig. 1. Main steps of text classification.

1. Data Gathering:

Since the dataset is an essential part of any machine learning project, classes must be precisely determined in accordance with the characteristics of the dataset. In this work, we have created a large dataset called "All-In-One Dataset" by combining four well-known and online available mask detection datasets.

In more detail, “All-In-One Dataset” is a combination of four datasets containing two categories of images: with and without mask. The first dataset was created by Vijay Kumar [4], and the total number of images in the dataset is 5988.

The second dataset [5] was created by scraping images from Google and Jessica Li's CelebFace; it consists of almost 12K images.

The RMFD dataset [6] is the third dataset created by combining a Kaggle dataset with photos gathered via the Bing search API; this dataset has 4079 images.

Finally, the Face Mask Detection dataset(FMD) [7]. This dataset contains images of real people typically wearing face

mask, which makes such dataset very representative of the real life systems.

2. Data pre-processing:

Data preprocessing involves converting raw data into a format that computers and machine learning programs can read and evaluate. The images are read and preprocessed in this step using the feature-rich computer vision library "OpenCV." Additionally, the image array is sent to TensorFlow's "preprocess" functions to ensure that each pixel has a comparable data distribution.

3. Feature extraction:

This step focuses primarily on finding and extracting distinctive features from the input images. The array generated by the preprocessing stage will be first normalized to the range [-1 to 1]. The image is then represented by an extracted vector made up of 49152 features. Note that "All-In-One Dataset" samples are separated into Training and testing different categories, one for each.

4. System classifier:

A successful classification procedure requires a classifier that is eligible according to the needs. This study examines the effectiveness of two cutting-edge CNN-based mask detectors. In more detail, these models are:

VGG19: The term "VGG" refers to a Convolutional Neural Network that Simonyan and Zisserman proposed in 2014 at the University of Oxford. The 19-layered VGG19 version uses three fully connected layers for classification and 16 convolutional layers for feature extraction. Then, each convolution layer is followed by 5 MaxPooling layers and 1 SoftMax layer. The ImageNet dataset, which contains 14 million images divided into 1000 classes, was primarily used to pre-train VGG19. In this study, we created a new version of the mask detection VGG19 model presented in [8].

MobileNet-v2: Google introduced the first version of this model in the first quarter of 2018. It consists of 53 layers and is also trained on the ImageNet database. An enhanced version of MobileNetV1 called MobileNetV2 was created to provide the opportunity to build computationally low-powered deep learning models for mobile-related systems. MobileNetV2, compared to V1, is 35% faster and can be considered a more productive extraction tool while maintaining the same accuracy as its former prototype. This study used a new version of the mask detection MobileNetV2 model [9].

It is worth mentioning that the input images are resized to 224x224 for both used models.

5. Classification output:

Each facial input image in this step belongs to one of two predefined classes; the first class is a person wearing a face mask, while the second one is a person not wearing a mask. As a result, the models in use will anticipate one output for each input image.

III. Experiment and Results

With the use of "All-In-One dataset", this section examines the effectiveness of the two used models, and to discover the ideal and effect of hyperparameter optimizations.

In addition, the accuracy, precision, recall, and F1 score evaluation metrics were used to guarantee the results' reliability. Note that the 3-fold cross-validation method is also used in hyperparameter optimization.

In this experiment, we first evaluated the original version of both MobileNetV2 and VGG19 models. Results are shown in Table 1.

Table 1. The Accuracy, F1, Precision, and Recall for the original version of MobileNetV2 and VGG19.

Model	Accuracy	F1	Precision	Recall
MobileNet V2	0.917	0.917	0.917	0.917
VGG19	0.975	0.976	0.976	0.976

Next, selecting the ideal hyperparameters is one of the key elements influencing how well a CNN model performs. Therefore, based on our preliminary investigation, the Model's parameters and the expected values that will be tuned were selected. Hyperparameters optimization was used to identify the number of filters, kernel size, Dropout value and Dense rate amongst our models. The following table demonstrates the input values that were investigated for hyperparameters optimization.

Table 2: The investigated hyperparameters and values.

Parameter	Used Values			
Kernel size	(1,1)	(3,3)	(5,5)	(7,7)
Filters	32	64	96	128
Dense rate	128	64	32	16
Dropout	0.2	0.25	0.3	0.5

Based on the values shown in Table 2, the performance of both VGG19 and MobileNetV2 while using random search to find the best structure and component was investigated. The Accuracy, F1, Precision, and Recall for the tuned version of MobileNetV2 and VGG19 are shown in Table 3.

Table 3. The Accuracy, F1, Precision, and Recall for optimized version of MobileNetV2 and VGG19.

Model	Accuracy	F1	Precision	Recall
VGG19	0.9947	0.9947	0.9948	0.9947
MobileNetV2	0.9925	0.9925	0.9926	0.9925

Overall, by comparing the results demonstrated in Tables 1 and 3, it is obverse that using random search has significantly improved the performance of both used models. In other words, the findings of this study demonstrated that applying the hyperparameter optimization strategy enhanced the models' overall performance. In addition, for the two parts of the experiment, i.e., with and without hyperparameter optimization, VGG19 was able to outperform MobileNetV2 using all in one dataset. However, in our preliminary experiments, MobileNetV2 achieved better results when tested on smaller datasets like the four mentioned ones, and its performance was dropped on large datasets.

Last but not least, Table 4, shows the values of the investigated parameters that achieved best performance.

Table 4: The selected values for the investigated hyperparameters.

Model	Kernel size	Filters	Dropout	Dense rate
VGG19	(3,3)	32	0,2	16
MobileNetV2	(1,1)	32	0,25	32

IV. CONCLUSION

CNN architectures are being used in increasingly fascinating and diverse ways. CNN is mainly used in image recognition and any task involving pixel data processing. Because CNN is multi-layered, it gives more accurate results than a single-layered network.

It is important to note that building a CNN architecture requires a lot of time and effort, especially with the addition of selecting the model layers and tuning their parameters.

In conclusion, this study examined the effects of CNN hyperparameter optimizations using Random Search. This is done using two well-known CNN models, i.e., MobileNetV2 and VGG19 for image classification. As for the datasets, 4 datasets were used and combined into one dataset. The conducted experiments show that the hyperparameter optimization technique has proved to enhance the performance of these two models while giving VGG19 a push in results across all evaluation metrics used.

REFERENCES

[1] Liashchynskyi, P., & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: a big comparison for NAS. arXiv preprint arXiv:1912.06059.

[2] Li, L., & Talwalkar, A. (2020, August). Random search and reproducibility for neural architecture search. In Uncertainty in artificial intelligence (pp. 367-377). PMLR.

[3] Mellor, J., Turner, J., Storkey, A., & Crowley, E. J. (2021, July). Neural architecture search without training. In International Conference on Machine Learning (pp. 7588-7598). PMLR.

[4] VIJAY KUMAR, (2021) Face Mask Detection, [Online]. Available: https://www.kaggle.com/datasets/vijaykumar1799/face-mask-detection.

[5] ASHISH JANGRA, (2021) Face Mask Detection~12K Images Dataset, [Online]. Available: https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset

[6] Real World Fasked Face Recognition Dataset (RMFRD), (2021) [Online]. Available: https://www.kaggle.com/datasets/muhammeddalkran/masked-facerecognition.

[7] Balaji S, (2021) Face-Mask-Detection, [Online]. Available: https://github.com/balajisrinivas/Face-Mask-Detection/tree/master/dataset/ with_mask

[8] Teboulbi S, Messaoud S, Hajjaji MA, Mtibaa A., "Face Mask Classification Based on Deep Learning Framework," InAdvanced Practical Approaches to Web Mining Techniques and Application, pp. 175-188, IGI Global, 2022.

[9] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC., "Mobilenetv2: Inverted residuals and linear bottlenecks," InProceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510-4520, 2018.